Revolutionizing Developer Experience With Automated Platform Engineering and DevOps

Table of Contents

The Challenge

- The software development landscape is rapidly evolving, putting immense pressure on development teams to deliver high-quality applications at an unprecedented pace
- Developers often spend unreasonable time doing undifferentiated heavy lifting rather than focussing on application code

  Example: A Java Microservice running on Cloud Kubernetes: Create Github repos, Write builds scripts, Run ci/cd pipelines, Run Terraform scripts to provision infrastructure, create docker files and kubernetes deployment Manifests
- General Developer Trend: Clones other Repositories and Tweak Names/Code. More Error prone, Repeated Iterations, Frustration
- This hinders developer productivity taking away their focus away from core app development. Can be more challenging for new developers who joined recently searching for information

The Solution

How soon can i get a quick environment to develop, deploy and test code right away **???**

Platform engineering, facilitated by Internal Developer Portals (IDPs), presents a strategic approach to alleviating these challenges.

IDPs provide pre-opinionated and pre-configured tools and environments, freeing developers to concentrate on their core responsibilities.

What is Platform Engineering

Primary goal is to lessen the cognitive burden on developers by offering automated processes and reliable tools

Design, development, and management of a combination of Software and Hardware frameworks that provide foundation for developing, running, and managing applications across the organization

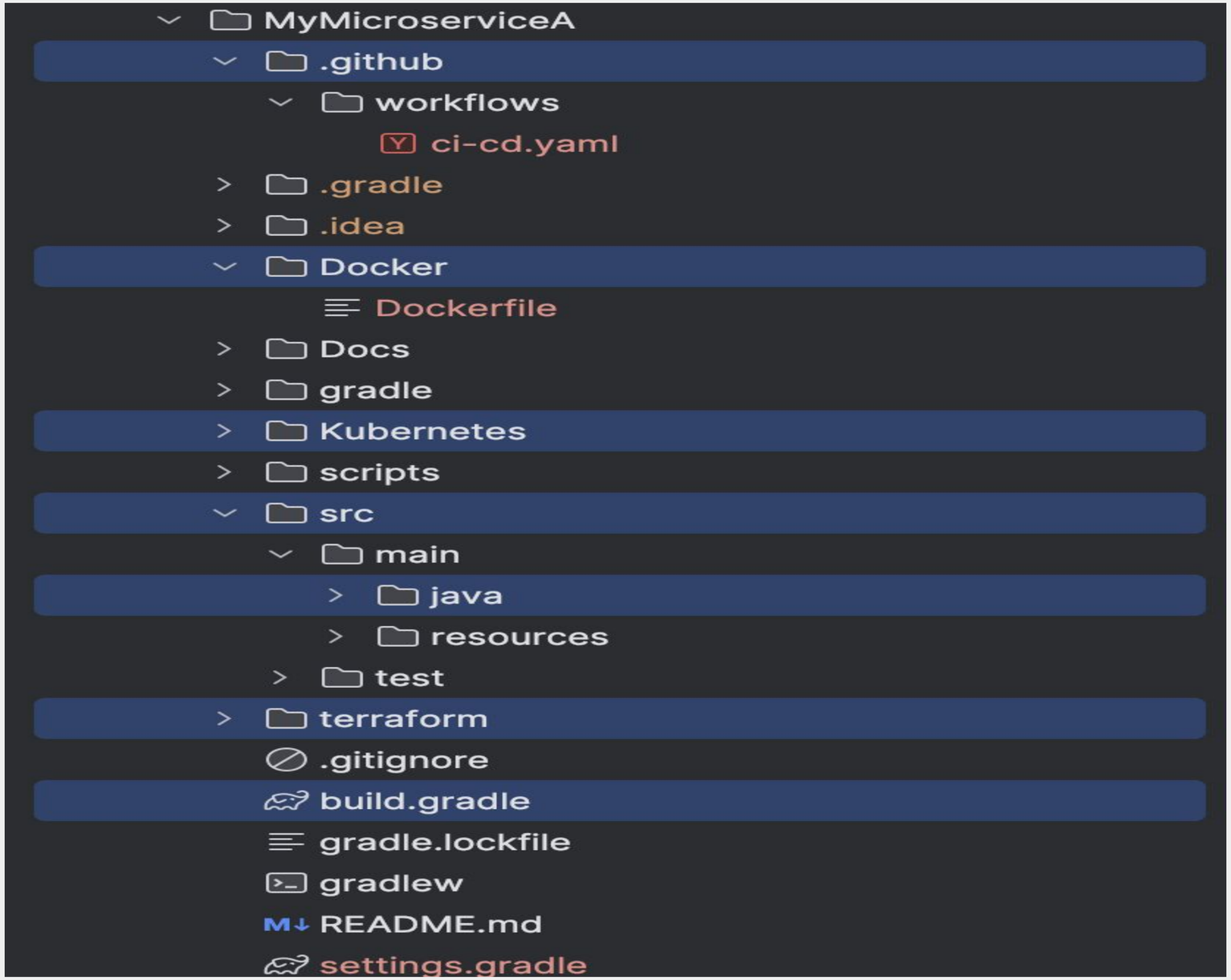
Key Aspects:

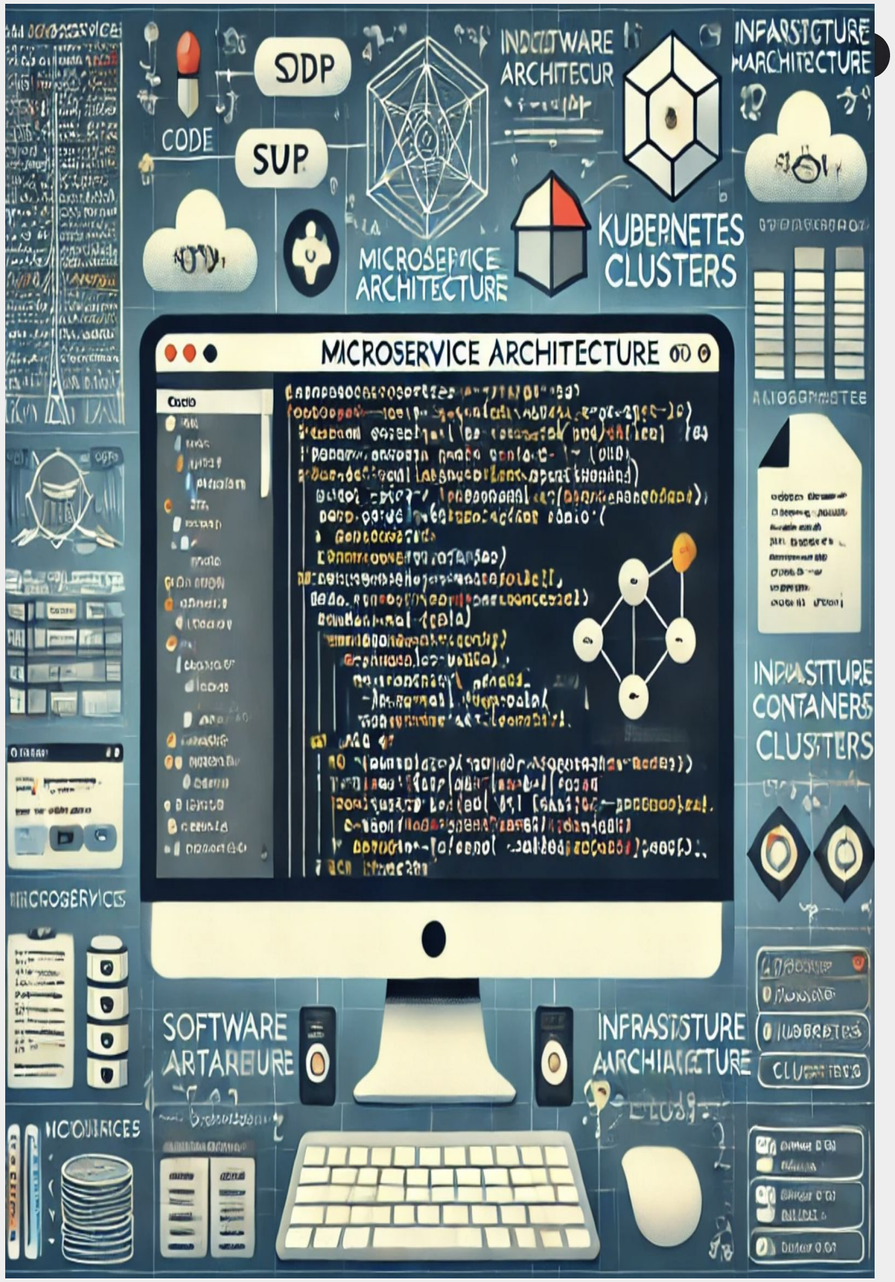
- **Development Platforms:** Auto provisioning of development environments such as code repositories, build scripts, tests, skeleton code
- **Deployment pipelines**: Automated ci/cd pipelines to release software any time
  Eg: github workflows, Jenkins pipeline scripts etc ..
- **Infrastructure Provisioning:** Automated scripts to provision infrastructure
  Eg:Terraform, Cloud Formation etc ..
- **Interoperability and Integration**: Makes it easy for New technology integrations and rollouts

Internal Developer Portals & Devops

- Interface between Developer and Platform
- Self-serve centralized hub for organization’s software development activities through an intuitive web portal
- Developers can provision fully configured environments for new projects by using ready-made templates
- Login to IDP, choose template of your choice . **(** Java, Node.Js, Python etc.) and provide inputs(Project Name, Project Team, Project email etc …). Get all necessary setups within minutes to start coding right away
- **Example**: A Github Repository created for a Java Microservice with : A Working Hello world skeleton Java code and tests, A github action workflow file for ci/cd , A Dockerfile, A Terraform file executed on repository creation that provisioned a Dev/Test/Prod Kubernetes Namespaces and an Artifactory repository to store artifacts
- Templates Owned and Maintained by **Devops** (Infrastructure as code, ci/cd, Docker)

Sample Repo with IDP





IDP Benefits

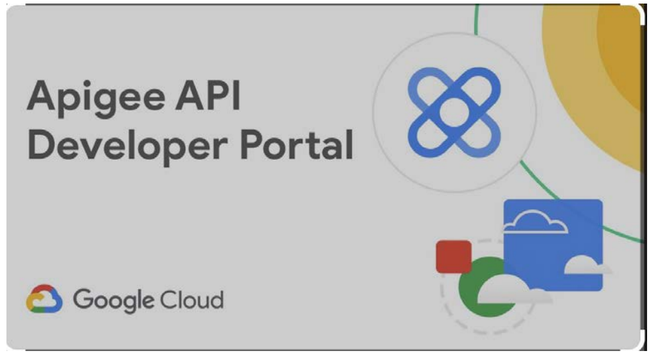
- **Standardization and Best Practises**: Ensures all teams follow the same best practices. More consistent and maintainable high quality code
- **Easy Onboarding**: One stop shop for all organizational resources. New developers can quickly learn technology stack and start contributing to projects right away
- **Developer Productivity Boost:** Centralize access to tools, documentation and API enable developers to focus more on coding and innovation and less of administrative tasks
- **Resource Management**: Centralized Monitoring and management of services and tools makes it easier for Platform/Devops/Infrastructure teams to manage updates, monitor usage, optimize resources across organization
- **Cost Management:** By improving developer efficiency, standardizing tools, and reducing administrative overhead, IDPs can lead to significant cost savings(Infrastructure/Software ROI) , particularly for large organizations

Platform Engineering + IDPs

- Goes hand in hand
- Seamless integration of backend systems with front-end accessibility
- Achieve more with less overhead across multiple disparate systems.
- Platform engineering creates the systems and tools, IDPs make them accessible and user-friendly for developers.
- A dual approach for robust application and infrastructure across the organization
- As organizations grow, platform can scale and evolve the infrastructure to meet increasing demands, and IDPs can be adapted to include new tools, resources, and support mechanisms.
- Automating routine tasks and Streamlining processes lead to quicker time-to-market and better-quality products.

IDP TOOLS


Backstage
Spotify

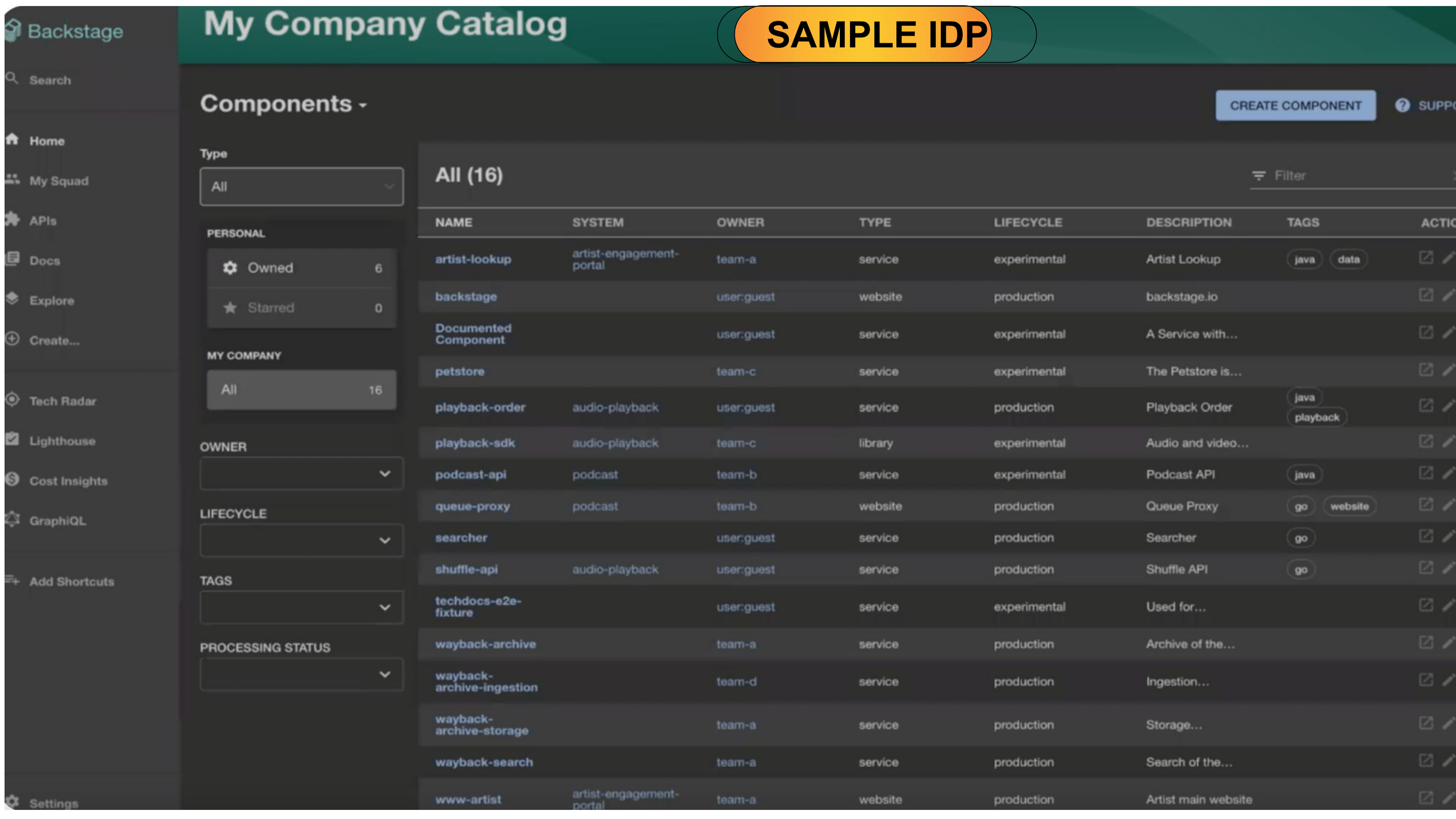

Apigee API
Developer Portal
Google Cloud


Swagger
Supported by SMARTBEAR


Red Hat
3scale API
Management


readme


Kong


Backstage
Search
Home
My Squad
APIs
Docs
Explore
Create...
Tech Radar
Lighthouse
Cost Insights
GraphiQL
Add Shortcuts
Settings
My Company Catalog
SAMPLE IDP
Components
CREATE COMPONENT
Type
All
PERSONAL
Owned 6
Starred 0
MY COMPANY
All 16
OWNER
LIFECYCLE
TAGS
PROCESSING STATUS
All (16)
Filter
NAME SYSTEM OWNER TYPE LIFECYCLE DESCRIPTION TAGS
artist-lookup artist-engagement-portal team-a service experimental Artist Lookup java data
backstage user:guest website production backstage.io
Documented Component user:guest service experimental A Service with…
petstore team-c service experimental The Petstore is…
playback-order audio-playback user:guest service production Playback Order java playback
playback-sdk audio-playback team-c library experimental Audio and video…
podcast-api podcast team-b service experimental Podcast API java
queue-proxy podcast team-b website production Queue Proxy go website
searcher user:guest service production Searcher go
shuffle-api audio-playback user:guest service production Shuffle API go
techdocs-e2e-fixture user:guest service experimental Used for…
wayback-archive team-a service production Archive of the…
wayback-archive-ingestion team-d service production Ingestion…
wayback-archive-storage team-a service production Storage…
wayback-search team-a service production Search of the…
www-artist artist-engagement-portal team-a website production Artist main website

localhost:3000/create?filters%5Bkind%5D=template&filters%5Buser%5D=all

Create a New Component

Create new software components using standard templates

Sample Templates with IDP

Available Templates

REGISTER EXIST

Search

PERSONAL

Starred 0

MY COMPANY

All 3

CATEGORIES

TAGS

Templates

documentation

Documentation Template

DESCRIPTION

Create a new standalone documentation project

OWNER

backstage/techdocs-core

TAGS

recommended techdocs mkdocs

CHOOSE

website

React SSR Template

DESCRIPTION

Create a website powered with Next.js

OWNER

web@example.com

TAGS

recommended react

CHOOSE

service

Spring Boot gRPC Service

DESCRIPTION

Create a simple microservice using gRPC and Spring Boot Java

OWNER

service@example.com

TAGS

recommended java grpc

CHOOSE

Metrics

Project Type	Number of Developers	Man-Hours Before	Man-Hours After	% Productivity Improvement
Web App Development	10	200	120	40%
Mobile Dev	15	300	150	50%
API Services	8	160	80	50%
Data Pipelines	5	100	60	40%
Security Tools	12	240	120	50%

Metrics Contd …

Project Type	Dev Satisfaction Before	Dev Satisfaction After	% Improvement
Web Apps	60	85	42%
Mobile Apps	55	80	45%
APIs	50	75	50%

Continuous Improvement is the Key

- **Regular Assessments:** Assess platform infrastructure and applications for performance, cost, scalability
- **Automate More processes:** Tedious manual tasks should be automated as much as possible to free developer burden
- **Feedback Loops**: Regularly collecting and analyzing feedback from developers to identify pain points and areas for enhancement in the portal
- **Updating Documentation**: Keeping the documentation up-to-date with the latest API changes, best practices, and tutorials to ensure it remains relevant and useful.
- **Tool Integration**: Continuously evaluating and integrating new tools that can enhance developer productivity and efficiency.
- **Performance Monitoring**: Implementing and improving monitoring tools to ensure the portal remains fast and reliable.

Conclusion

In the rapidly evolving software development landscape, platform engineering and Internal Developer Portals (IDPs) have emerged as a game-changing solution to enhance developer experience and operational efficiency. By automating infrastructure management, providing pre-configured tools and environments, and streamlining processes, developers are empowered to focus more on innovation

Thank You